



**Australian Government**

**Department of Defence**

Defence Science and  
Technology Organisation

# **Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Developer Guide**

Egon Kuster

DSTO-GD-0377

**DISTRIBUTION STATEMENT A**  
Approved for Public Release  
Distribution Unlimited

**Reproduced From  
Best Available Copy**

**20040412 031**



Australian Government  
Department of Defence  
Defence Science and  
Technology Organisation

# Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Developer Guide

*Egon Kuster*

Command and Control Division  
Information Sciences Laboratory

DSTO-GD-0377

## ABSTRACT

This document has been created for the developer of the WDCT system. It contains documentation of the WDCT architecture and components that are part of the system. To support this document is the WDCT Developer CD-ROM, which contains all the WDCT source files and support files that are required in the development or alteration of the system. The structure of the database and the data contained within is explained. All Java Class files are discussed as to their function within the WDCT. Each JSP used is described as to the functionality that it provides. Also contained is a brief discussion of what new functionality that could be added.

## RELEASE LIMITATION

*Approved for public release*

AQ F04-07-0394

*Published by*

*DSTO  
Information Sciences Laboratory  
PO Box 1500  
Edinburgh South Australia 5111 Australia*

*Telephone (08) 8259 5555  
Fax: (08) 8259 6567*

*© Commonwealth of Australia 2003  
AR-012-920  
September 2003*

**APPROVED FOR PUBLIC RELEASE**

# Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Developer Guide

## Executive Summary

This document is a supplement to the WDCT User Guide and WDCT Installation Guide. It has been created to support future developers of the WDCT system so that they can have a better understanding of its inner workings. This document has been produced to act as a guide for development and as a quick reference. The document is broken up into:

- **Technology Overview** – Contains an overview of the software used within the system and the capabilities that each of these technologies provides. Also listed is a set of recommended software for the development and deployment of the WDCT.
- **Developer CD-ROM** – This section describes the contents of the WDCT Developer CD-ROM. If you are trying to find the location of a resource read this section to understand the structure and storage of the resource files used.
- **Developer Installation Guide** – Instructions on how to install each of the tools and applications required for conducting development on the WDCT.
- **WDCT Servlet Structure Basics** – This section provides a basic guide on how Java Servlets are developed and how they operate within the context of the WDCT system. This is extremely useful if the developer has never programmed Java Servlets before.
- **WDCT Classes** – Each Java class used in the WDCT system is explained. Outlined are their session variables, input parameters and the functionality each Java class provides.
- **WDCT Java Server Pages (JSP)** – The JSP pages used within the WDCT provide some of the user interface. This section describes the functionality of every JSP used in the WDCT.
- **WDCT Database Schema** – As the WDCT relies heavily on the use of an Oracle Database this section outlines all the tables contained in the database schema. For each table all the columns are described along with what the table is used for.
- **Future Directions** – This section outlines a number of tasks or additional functionality that are currently not part of the WDCT system but could be developed in the future.

## Authors

### Egon Kuster Command and Control Division

*Egon Kuster has worked as a web developer for the past six years. Current work is as the Australian Coalition Theatre Logistics (CTL) Technical Manager overseeing the research and development into the CTL demonstrator, which is being jointly developed between Australia and America. Egon has also been the South Australian node manager of the Combined Federated Battle Laboratory network (CFBLnet), which is a large multinational secret demonstration network. Primary skills include the development of large, complex distributed solutions based on web technologies. Egon has experience in creating software solutions for information management, distributed data access, Enterprise Information Portals (EIP) and web service communication.*

---

---

# Contents

1. PURPOSE OF THIS DOCUMENT.....	1
2. TECHNOLOGY OVERVIEW.....	1
2.1 Recommended Software and Development Kits.....	2
3. DEVELOPER VERSION WDCT CD-ROM.....	3
3.1 Directory Structure.....	3
3.1.1 Docs Directory .....	3
3.1.2 GraphicsSource Directory .....	3
3.1.3 IE55 Directory .....	4
3.1.4 Jakarta Directory .....	4
3.1.5 Java Directory .....	4
3.1.6 Javadoc Directory .....	5
3.1.7 JavaSource Directory .....	5
3.1.8 Oracle Directory .....	5
3.1.9 WDCT Directory.....	5
3.1.10 WinZip Directory .....	5
4. DEVELOPER INSTALLATION GUIDE.....	6
4.1 The Operating System.....	6
4.2 Developer Tools .....	6
4.2.1 Java 2 SDK SE .....	6
4.2.2 Database .....	7
4.2.3 Editing Tools.....	7
4.3 Files For Development .....	8
4.4 Getting Started.....	8
5. WDCT SERVLET STRUCTURE BASICS .....	12
5.1 Input .....	12
5.2 Initial Input Handling.....	13
5.3 Servlet Logic.....	13
5.4 HTTP Session.....	13
5.5 DB.....	13
5.6 WDCT Components.....	13
5.7 Output .....	14
6. WDCT CLASSES .....	14
6.1 AdministrationArea.class .....	14
6.2 AdministratorLogin.class .....	14
6.3 Apostrophe.class.....	15
6.4 AssessmentAnalyticalReport.class .....	15
6.5 AssessmentStatusReport.class.....	16
6.6 ConfigurationAdmin.class .....	16
6.7 ConsentForm.class .....	17
6.8 ContactInfo.class .....	17
6.9 CountDown.class .....	18
6.10 CreateNvivoDocuments.class .....	18
6.11 DBConnection.class .....	18

6.12	DemonstrationAdmin.class.....	18
6.13	Demonstrator.class.....	19
6.14	DemonstratorLog.class.....	19
6.15	DemonstratorLogReport.class .....	20
6.16	destroySession.class.....	21
6.17	EnterProfile.class.....	21
6.18	LessonsLearned.class.....	22
6.19	LessonsLearnedReport.class .....	22
6.20	NetworkAdmin.class .....	22
6.21	NetworkAdminLog.class .....	23
6.22	NetworkPerformanceReport.class .....	23
6.23	QuestionAdmin.class .....	24
6.24	Questions.class .....	25
6.25	UsersReport.class .....	26
6.26	Warfighter.class .....	27
6.27	WarfighterLog.class .....	27
6.28	WarfighterLogin.class .....	28
6.29	warfighter_split.class.....	28
<b>7.</b>	<b>WDCT JAVA SERVER PAGES (JSP).....</b>	<b>29</b>
7.1	administrator_login.jsp.....	29
7.2	administrator_login_error.jsp.....	29
7.3	db_error.jsp .....	29
7.4	index.jsp.....	29
7.5	invalid_session.jsp.....	29
7.6	warfighter_login.jsp .....	29
<b>8.</b>	<b>WDCT DATABASE SCHEMA.....</b>	<b>30</b>
8.1	<b>ADMINISTRATORS Table .....</b>	<b>30</b>
8.1.1	Columns.....	30
8.1.2	Description .....	30
8.2	<b>ANSWERS Table.....</b>	<b>30</b>
8.2.1	Columns.....	30
8.2.2	Description .....	31
8.3	<b>DEFENCE_SPONSOR Table.....</b>	<b>31</b>
8.3.1	Columns.....	31
8.3.2	Description .....	31
8.4	<b>DEMONSTRATIONS Table .....</b>	<b>31</b>
8.4.1	Columns.....	31
8.4.2	Description .....	31
8.5	<b>DEMONSTRATION_SPONSOR Table.....</b>	<b>32</b>
8.5.1	Columns.....	32
8.5.2	Description .....	32
8.6	<b>DEMONSTRATOR_LOG Table.....</b>	<b>32</b>
8.6.1	Columns.....	32
8.6.2	Description .....	32
8.7	<b>DEMO_LOCATIONS Table.....</b>	<b>33</b>
8.7.1	Columns.....	33
8.7.2	Description .....	33
8.8	<b>JWID_ADMIN Table.....</b>	<b>33</b>
8.8.1	Columns.....	33

8.8.2	Description .....	34
<b>8.9</b>	<b>LESSONSLEARNT Table.....</b>	<b>34</b>
8.9.1	Columns.....	34
8.9.2	Description .....	34
<b>8.10</b>	<b>LOCATION_LOV Table.....</b>	<b>34</b>
8.10.1	Columns .....	34
8.10.2	Description .....	34
<b>8.11</b>	<b>MULTIPLE_CHOICE Table.....</b>	<b>34</b>
8.11.1	Columns .....	34
8.11.2	Description .....	35
<b>8.12</b>	<b>NETWORK_ADMIN_LOG Table.....</b>	<b>35</b>
8.12.1	Columns .....	35
8.12.2	Description .....	35
<b>8.13</b>	<b>PROFILES Table.....</b>	<b>35</b>
8.13.1	Columns .....	35
8.13.2	Description .....	36
<b>8.14</b>	<b>QUESTIONS Table.....</b>	<b>36</b>
8.14.1	Columns .....	36
8.14.2	Description .....	36
<b>8.15</b>	<b>USERS Table.....</b>	<b>36</b>
8.15.1	Columns .....	36
8.15.2	Description .....	37
<b>8.16</b>	<b>WARFIGHTER_LOG Table.....</b>	<b>37</b>
8.16.1	Columns .....	37
8.16.2	Description .....	37
<b>8.17</b>	<b>WARFIGHTER_SPONSOR Table.....</b>	<b>37</b>
8.17.1	Columns .....	37
8.17.2	Description .....	37
<b>9.</b>	<b>FUTURE DIRECTIONS.....</b>	<b>38</b>
9.1	Database Connection Pool.....	38
9.2	User Input Checking and Verification.....	38
9.3	Browser Compatibility .....	39
9.4	Role Based Administrator system.....	39
9.5	More Sophisticated Question System.....	39
9.6	Export Functionality .....	40
<b>10.</b>	<b>CONCLUSION.....</b>	<b>40</b>

## **Abbreviations**

CSS	Cascading Style Sheets
DB	Database
DHTML	Dynamic HyperText Markup Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
JDBC	Java Database Connectivity
JDK	Java Development Kit
JSP	Java Server Pages
JWID	Joint Warrior Interoperability Demonstration
MS	Microsoft
OS	Operating System
RDBMS	Relational DataBase Management System
SDK	Software Development Kit
SE	Standard Edition
SQL	Structured Query Language
URI	Universal Resource Identifier
URL	Universal Resource Locator
WDCT	Web Data Collection Tool

## 1. Purpose of this document

This document is intended for future developers of the WDCT system. It serves as a starting point for new developers and persons who are interested in how the WDCT system operates. It may also be used by system administrators who wish to customise their installation of the WDCT in a way that is beyond the scope of the installation guide.

The document is structured to be progressively detailed. This way the reader can gain a gradual understanding of the WDCT system and the issues and requirements for developing this system further. Once the first few chapters of this document are read, the developer may choose to focus on a particular aspect of the system and thus will not need to read this document in full, if only one section of the WDCT is going to be altered.

Unlike the Installation Guide, this document is not aimed at a general computer user. The reader is expected to have knowledge of their target development platform. Further they should be familiar with Java and relational databases. This knowledge is essential for full understanding of this document and to conduct development or alterations on the WDCT system.

## 2. Technology Overview

The JVID WDCT system uses the following technologies:

- Java 2 Platform
- Java Servlets (Servlets)
- Java Server Pages (JSP)
- Jakarta Tomcat servlet and JSP container
- Oracle Relational DataBase Management System (RDBMS)
- HyperText Markup Language (HTML) and Dynamic HTML (DHTML)
- JavaScript
- Cascading Style Sheets (CSS)

The WDCT system is a web based, multi-user application. It can be distributed and multiple instances of the software can be run simultaneously, potentially on the same machine. The key functionality of the system is in the form of Servlets and JSPs deployed within the Jakarta Tomcat or similar Java servlet container. The Java servlet container is used within a standalone mode to provide content to its users, although it can be deployed in conjunction with another HTTP (HyperText Transfer Protocol) Server, like Apache. This document will only discuss using the servlet container within standalone mode. For information about connecting the servlet container to a HTTP server consult the servlet container's documentation. The Servlets and JSPs used in the WDCT collect data from users and store it within an Oracle DB (DataBase). Users

interact with the system via a standard web browser; the recommended browser is Microsoft Internet Explorer version 5.5 although other browsers are capable of accessing most of the WDCT functionality. Some parts of the system can produce unusual behavior if Internet Explorer is not used.

As Java Servlets and JSPs make up the bulk of the system it is recommended to become familiar with programming with these two technologies. There are two O'Reilly Books that are highly recommended:

Java Servlet Programming, 2<sup>nd</sup> Edition [1]

JavaServer Pages, 2<sup>nd</sup> Edition [2]

## 2.1 Recommended Software and Development Kits

Recommended versions of software and APIs are:

- Java 2 – SDK 1.3.1 Standard Edition
- Java Servlet API 2.2
- Java Server Pages API 1.1
- Jakarta Tomcat 3.2.3
- Oracle 8i 8.1.7
- Microsoft Internet Explorer 5.5

For development purposes a developer may also require an editor capable of editing Java, XML, JSP and other text files. It is highly recommended that a suitable Java development package be installed such as Sun's Forte for Java, although a plain text editor will suffice.

It is also highly recommended that a good HTML editor is used for editing the HTML content in WDCT. Macromedia Dreamweaver is an excellent choice for editing and designing new HTML based content. Dreamweaver 4 can also be used to edit the JSP files. This will greatly decrease the development time as the JSP source code can get quite involved and become cumbersome to edit using a pure text editor. Do not use any GUI (Graphical User Interface) based HTML editor that will modify already existing code as this can break already written HTML and JavaScript. Microsoft Frontpage must not be used, as it will rewrite HTML breaking the embedded JavaScript and corrupting the source files.

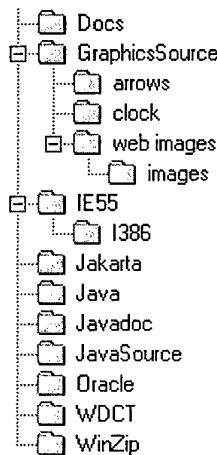
If any of the graphics files used within the system are to be edited it is recommended to use Adobe Photoshop. Many source files for the graphics in WDCT system are saved in the ".psd" Photoshop file format. Many other graphics packages available on the market also support this format, so it is not necessary to only use Adobe Photoshop.

### 3. Developer version WDCT CD-ROM

The developer version of the WDCT CD-ROM consists of the standard contents from the WDCT distribution CD-ROM with the addition of source code, graphics files and API documents for the WDCT system.

#### 3.1 Directory Structure

The directory structure of the WDCT CD-ROM is shown in Figure 1. The most notable addition to the standard distribution of the WDCT system is the “GraphicsSource” and the “JavaSource” directories.



*Figure 1 - Directory Structure of the Developer WDCT CD*

##### 3.1.1 Docs Directory

Contains the documentation for the WDCT. This directory contains the WDCT User Guide, WDCT Installation Guide and WDCT Developer Guide (this document). If you have any questions about how to use, install, or alter the WDCT the first place you should look is in the documents contained within this directory.

##### 3.1.2 GraphicsSource Directory

Contains the source graphics for the images used in the WDCT. The main graphics source files have been saved using the Adobe Photoshop file format that can be read by many good quality graphics editors.

The WebImages subdirectory contains the bitmaps used in the current version of the WDCT. This is meant to serve as a quick reference only and a duplicate of these files is found in the deployed context.

The clock subdirectory contains the graphics for creating the session timeout clock, which is displayed in the top right-hand corner of most pages throughout this site.

The arrows subdirectory contains images that are used in the creation of the arrow images throughout the WDCT site.

### 3.1.3 IE55 Directory

Contained within this directory is a full installation of Microsoft Internet Explorer 5.5 for the win32 platform. This is the recommended browser for using the WDCT. Install this browser on all machines that are to access the WDCT. For installation instructions refer to the "README.TXT" file contained within this directory. Newer versions of Internet Explorer can also be downloaded from Microsofts home site.

Microsoft Home Site: <HTTP://www.microsoft.com.au/>

### 3.1.4 Jakarta Directory

Contains the files related to the Jakarta Tomcat project. This includes a distribution of the Jakarta Tomcat servlet container as well as the associated servlet API. The version included on the CD of the current WDCT is version 3.2.3. This is the recommended version for development although maintenance versions of Jakarta Tomcat 3.2.x should be 100% compatible. The WDCT system was also tested on versions lower than 3.2.3, however these had serious security issues, which were addressed in later versions. For a full account of the changes consult the Jakarta Tomcat 3.2.x documentation. More information on Tomcat can be found on its home site.

Jakarta Tomcat Home Site: <HTTP://jakarta.apache.org/tomcat/>

### 3.1.5 Java Directory

Contains a Windows (win32) distribution of the Java 2 SDK version 1.3.1. This is an essential part for developing the Java components for the WDCT. Although the win32 version of the Java SDK is included on the CD-ROM it is not imperative to develop on the win32 platform as servlets built on other platforms will also work on the win32 platform because Java servlets are platform independent. For example, it is possible to run Jakarta Tomcat serving the WDCT and Oracle DB on a Unix based platform, such as Linux or on a win32 machine using the same servlets and JSPs.

To acquire a non-win32 version of the Java 2 SDK visit the sun website. Extra Java documentation can also be found at Sun's Java site.

Sun Java Home Site: <HTTP://java.sun.com/j2se/>

### 3.1.6 Javadoc Directory

Contains Javadoc-generated documentation for all classes used within the WDCT. Javadoc is a tool for automating the creation of Java class documentation. It creates a HTML reference for Java classes based on comments entered by authors into the source code. The generated Javadoc web site is an invaluable tool for quickly accessing the structure of each class and its intended function. To access the Javadoc html simply direct your web browser to “[CD-DRIVE]:/Javadoc/index.html”.

### 3.1.7 JavaSource Directory

Contains all the Java files used in the WDCT. These files are the source code for the servlets and supporting Java classes.

### 3.1.8 Oracle Directory

Contains the database SQL (Structured Query Language) script for creating all database objects required by the WDCT. This configuration script has only been tested with Oracle, however other SQL compatible database systems, which have a similar feature list as Oracle should be able to make use of the file without major alterations, please consult your database manuals for compatibility.

### 3.1.9 WDCT Directory

Contains the deployable WDCT context. The term “context” is used to define the file and directory structure needed to deploy the WDCT software within the Jakarta Tomcat servlet container. The structure of the context is self-contained and requires little or no external files to function correctly depending on the desired configuration. Currently the context is distributed as a ZIP archive, as it may sometimes require alterations if the user chooses non-standard configuration options. This means that to deploy WDCT the user has to decompress the context. To simplify the deployment it is possible to create a WAR file using Java tools, which completely encapsulates the context, however configuration of WDCT would be more complicated.

### 3.1.10 WinZip Directory

Contains a Win32 distribution of the WinZip software. This software allows for decompression and creation of the ZIP files. The compressed zip file format is used throughout the WDCT CDs to archive other files and directory structures. The key advantage for using ZIP archives to store files is that it can be used to preserve directory structures for transport and distribution. There are a number of other compression and archival programs that are capable of decompression and creation of ZIP files.

## 4. Developer Installation Guide

This section describes how to install all the relevant tools and files ready for the development or alteration of the functionality already found within the WDCT system.

### 4.1 The Operating System

The choice of an Operating System for developing WDCT is restricted to the availability of a Java 2 Standard Edition SDK. Currently there are three operating systems for which Sun provides Java 2 SDK. These are, Solaris, Linux and Microsoft Windows. Although the new MacOSX operating system also supports Java 2, this is not an official Sun product, instead it is a port based on the Solaris JDK 1.3. There are also a number of other ports available. The full list can be found at "<http://java.sun.com/cgi-bin/java-ports.cgi>".

This leaves a fairly broad selection pool for the developer. The three Sun-produced Java 2 SDKs are always the first to receive updates. Further it will be easier to receive Java 2 support for these SDKs, this is why they are recommended for developing the WDCT. The final choice should be based on the developer's preference.

It should be taken into consideration that Microsoft Internet Explorer 5.5 is the target client browser. This browser does not, at the time of writing, run on any platform other than MS Windows. There is a beta version of this software for the Solaris platform, and a Mac version. However these may not be suitable for development purposes as they are not as complete in functionality. Therefore, unless the developer is going to run more than one machine, MS Windows 2000 or NT is the suggested OS (Operating System) for development.

### 4.2 Developer Tools

The developer tools required for development may vary depending on what area of the WDCT is relevant. For instance if only graphical development is taking place and no new functionality is being introduced, an HTML editor and graphics editor may suffice. For most development work, graphics, HTML, and code will need to be edited and therefore all development tools will be needed.

#### 4.2.1 Java 2 SDK SE

The primary tool for developing the WDCT is the Java 2 SDK. The Standard Edition of the SDK version 1.3.1 is used as specified in 2. Obtain this or a newer version of software suitable for your OS and install it.

It is possible to have more than one version of the SDK installed on one computer. This could be useful when developing a WDCT version for a specific SDK implementation

concurrently with a new technology test for example. When installing multiple versions of the SDK the developer needs to keep track of the relevant environment variables on their systems to ensure that the correct SDK installation is referenced when needed. This is done with the "JAVA\_HOME" and "PATH" environment variables. These should contain the correct directory paths pointing to the desired Java installation.

#### 4.2.2 Database

The database used for this version of WDCT is Oracle. The version that the WDCT was developed and tested on was Oracle8i.1.7 . Because WDCT uses JDBC (Java Database Connectivity) to perform all its database interaction the new versions of Oracle should not have any impact on the WDCT functionality. JDBC provides a neutral method of searching and accessing data contained in any database. Many major database manufacturers provide JDBC support.

Ideally, Oracle should be installed on a different machine to that used for main development. If possible, the developer should seek to use an already installed Oracle database that is managed by a database administrator. This way the complication of administering an Oracle server, quite a task by itself, can be delegated to the DB administrator resulting in a more efficient use of the WDCT developer's time.

The WDCT does use some Oracle specific SQL statements within the servlets. For this reason the WDCT can't be used with a different database without slight modifications. Should adaptation to a different database be required, the developer should ensure that the database has JDBC drivers and that it is SQL compatible. This will minimise the effort required to adapt the WDCT for use with the new database.

#### 4.2.3 Editing Tools

The main editing tool required by a WDCT developer is a text editor. It is highly recommended that an IDE (Integrated Development Environment) that supports Java and its child technologies be used. The recommended IDE is Sun's Forte for Java. This IDE supports ALL technologies used by the WDCT and is in continuous development. Forte for Java is also available as a Community Edition or Enterprise Edition. The community edition is distributed free of charge and supports all the technologies used by WDCT. Forte for Java is a Java application, which means that it is available on any platform that Java 2 runs on. This is gives it an advantage over other IDEs that are not available on multiple platforms.

There are many other Java IDEs available on the market. It is up to the developer to choose an IDE that they are familiar with and is capable of editing JSP and Java servlets.

A graphic editor will also be required. The recommended tool for this is Adobe Photoshop as many source files for the graphics in WDCT system are saved in the .psd Photoshop file format. Other graphic tools may also be necessary to create animations and vector graphics. If new graphics are to be created during the development effort then a vector based graphics editor is also recommended. Photoshop is a raster based image editor, which means that it will edit pixel-based images. Vector-based images are created using lines, circles, squares, polygons and curves to create the complete image. Generally when creating new graphics it is best to develop the initial concept using the vector-based graphics tool and import the created graphic into a raster-based graphics tool for minor changes and compositing.

Another editing tool that will be useful for developing WDCT is a HTML editor. Since the output of the WDCT system is almost entirely HTML based it is useful to have a tool that enables rapid creation of web content. A good tool to perform these tasks is Macromedia Dreamweaver. In fact other Macromedia products could also be used to further improve content creation speed.

### **4.3 Files For Development**

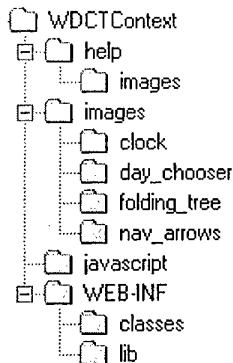
The files needed for development are the Java source of servlets, the graphic source of the site images and all the files contained in the WDCT context.

The Java source files need to be loaded up in the Java editor/IDE of your choice. Most of these files require that the Oracle JDBC driver and the Servlet classes to be in your system's classpath to compile.

The JSP and HTML files that WDCT uses are located in the WDCT context itself. The reason for this is that they do not need to be compiled or processed in any way for distribution. Thus it does not make sense to separate them out of the context. Further, placing these files outside of the context directory structure will invalidate the static links to other resources.

### **4.4 Getting Started**

To start the development work on WDCT, the developer(s) should make themselves familiar with the structure of the WDCT context. Figure 2 shows the directory structure.



*Figure 2 - WDCT Context Structure*

The root directory contains all JSP files that WDCT uses. It also contains static HTML files that are used instead of JSP when no dynamic content is needed. An example of a static page is `PrintControl.html`, which as the name suggests is used to control the printing of WDCT output. The main file of the system, `index.jsp` is also located in the root directory. This file's output is the first page that a WDCT user sees when entering the WDCT system. For summaries of the JSP files consult section 7.

The `help` directory contains all the help files for the system. The help files are static and thus are deployed as plain HTML files. The `help` directory has an `images` subdirectory containing the graphics used by help files. This structure separates the WDCT help system from the rest of the WDCT areas.

The `images` directory contains all graphics used in the WDCT except the help system. The `images` directory has several subdirectories containing images for specific WDCT functional areas. This structure was introduced purely for the convenience of the developer so that images can be tracked down easier as there are around 200 images in total.

The `Javascript` directory contains all Javascripts used in the WDCT system. The Javascript files are referenced in the HTML output of other WDCT components. They provide key functional aspects of the system, and most likely will need modification if major changes to WDCT are needed.

The `WEB-INF` directory is a special context directory. Unlike all the other directories, its contents are not available to the end users of WDCT. This is where the context configuration and programming logic is stored. The context configuration file `web.xml` is stored in the root of `WEB-INF`. The `classes` subdirectory contains ALL the compiled Java code. The servlet container, Jakarta Tomcat, executes these Java class files as needed. These Java classes contain Servlet code and supporting classes. The `lib` subdirectory contains all the libraries needed by the classes to function. Jakarta Tomcat will automatically make use of these libraries. The libraries can be placed in here in their original directory structure, or as a package in the JAR file format. Since Tomcat

3.2.3, ZIP file format is also supported for storing libraries. This is where the Oracle JDBC drivers are stored.

Once familiar with the structure of the context, the developer should deploy a context and run it to become familiar with the actual WDCT system as the user sees it. To do this Java 2, Oracle and Jakarta Tomcat need to be installed and the WDCT context placed in Tomcat's webapps directory. Traversing through the deployed and running WDCT the developer should take note of the URI in their browser to find out which class/JSP/HTML files produce the output they are looking at. Figure 3 on page 11 displays the complete site map, this is helpful to identify all the pages used within the WDCT. It is also useful to refer to section 5 for a summary of Java class functionality and the URL variables they use to communicate to other Java servlets and JSPs.

Now that the functional nature of the site has been explored the database schema used should be examined. This is crucial to understanding how data is stored in the system. The database schema is defined in the `create_db.sql` file in the Oracle directory on this CD. It is also summarised in section 8.

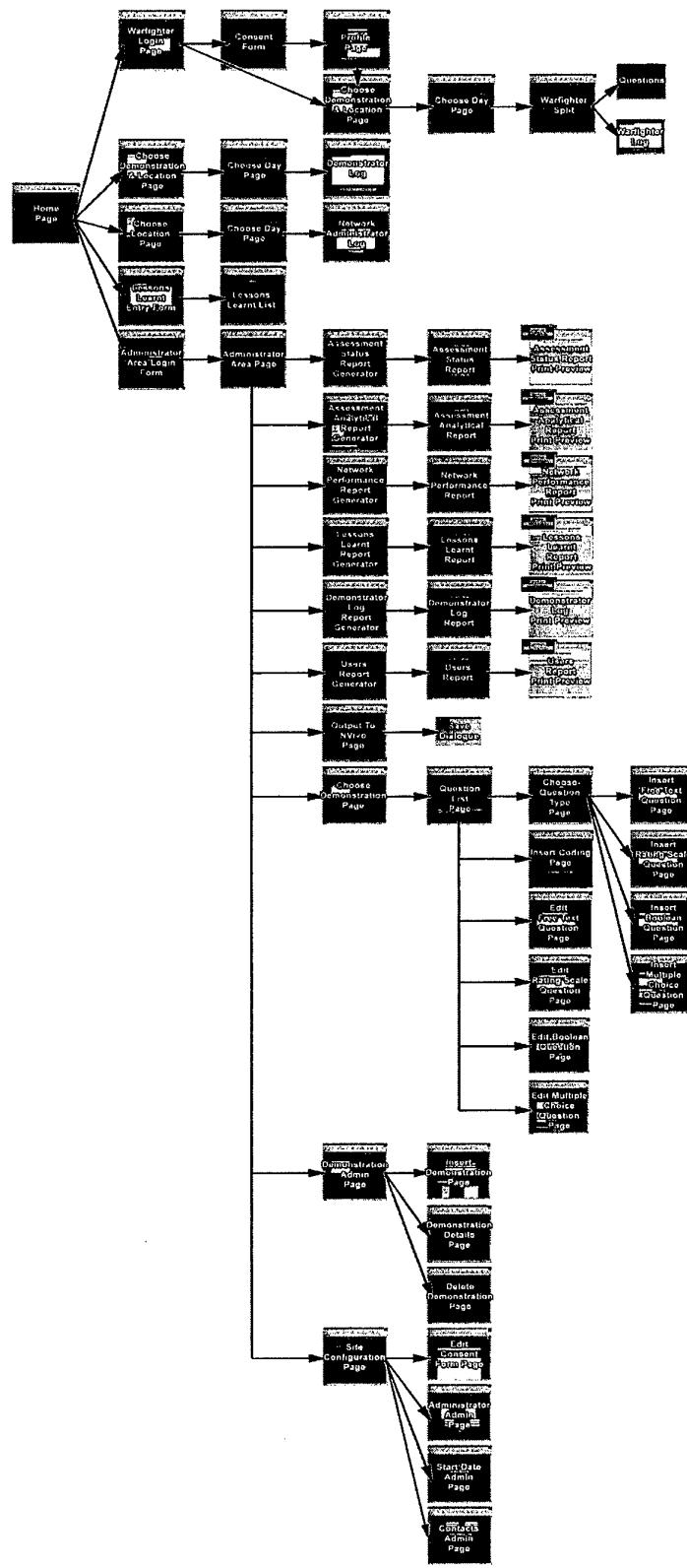


Figure 3 - WDCT Site Map

## 5. WDCT Servlet Structure Basics

Servlets form the key components of the WDCT. Most of the WDCT servlets follow a similar structure. This chapter is an introduction to this structure. For general servlet structure information please consult:

<http://java.sun.com/products/servlet/whitepaper.html>.

Figure 4 is a high level abstraction of the key aspects of the WDCT servlets. The figure shows the input and output as well as the processing done in most WDCT servlets.

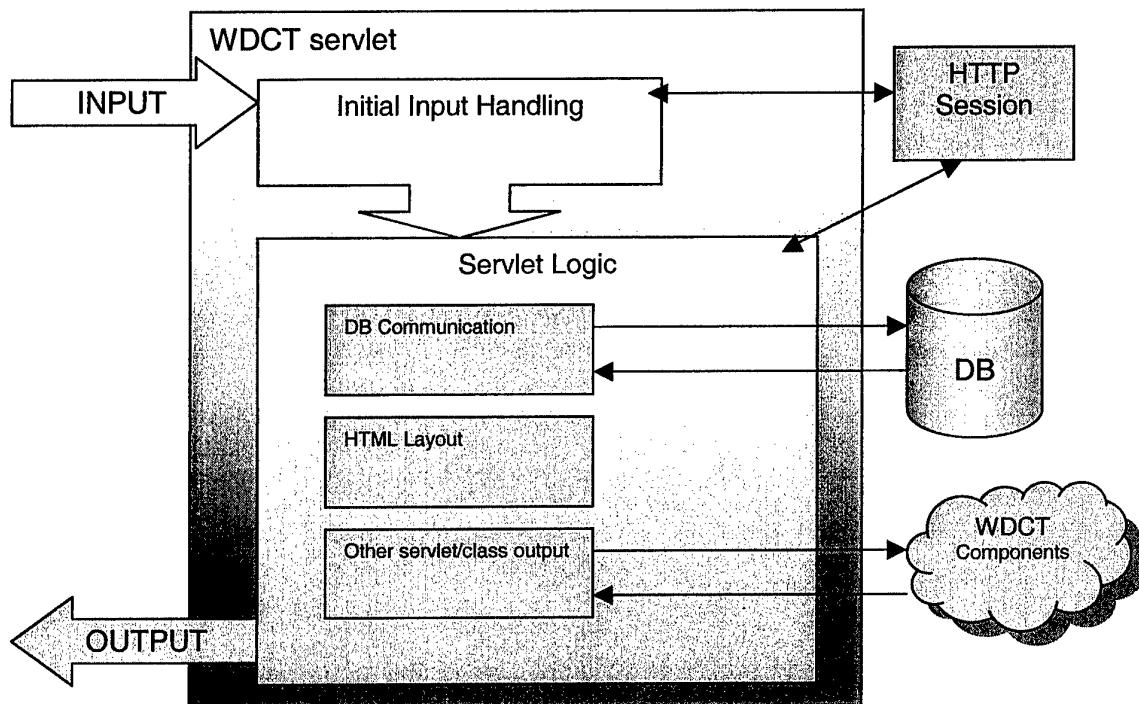


Figure 4 - Standard WDCT Servlet Structure

### 5.1 Input

This is usually a user HTTP request. In most cases this will have some parameters associated with it. It can also be a HTTP request from another WDCT component, for instance a redirect from one Servlet to another.

## 5.2 Initial Input Handling

Most servlets implement a service method or a doPost method. These methods are invoked when an input is received. Each servlet reads its input and checks for any input parameters. Based on the input parameters contents it then forwards the HTTP request onto all or a part of its logic methods.

The HTTP Session object for the user is also checked here, to further determine which logic should be invoked. In some servlets that do not have complicated tasks to perform, the servlet logic is also defined in this service or doPost method.

## 5.3 Servlet Logic

Defines what task the servlet performs. This can comprise of one or many methods. The usual functionality defined in WDCT servlets is that of a DB connection to store or retrieve data and producing HTML layout for presenting the DB data to the user. Additionally servlets often include the output of another servlet or class through servlet chaining. This can often be very useful to reduce code duplication and remove unnecessary instantiation of classes.

## 5.4 HTTP Session

This is a very important concept and the WDCT relies upon it heavily. Every time a user accesses the WDCT they are assigned a HTTP session object. This object stores state information about the user. In this way the progress of a user can be tracked throughout the WDCT enabling its functionality. Information like the demonstration the user selects when answering questions is stored within the HTTP session object. Without the session object the WDCT system will not operate correctly. The Servlet container handles most of the session management although there is some code in the index.jsp page, which displays the site home page, to nullify the session object for the site to operate cleanly.

## 5.5 DB

The database used by WDCT is Oracle, however it is possible to use other SQL databases with slight modifications to the system as a whole. The communication with the database software is accomplished through JDBC. For more information about JDBC refer to the online Java tutorial found at Sun's web site:

<http://java.sun.com/>

## 5.6 WDCT Components

The different components of the WDCT are the classes, servlets, JSPs, images, Javascript, cascading style sheets and configuration files.

## 5.7 Output

Is usually in HTML format for viewing in a web browser. Some servlets' output only part of a whole HTML page and is not intended for use in standalone form. For instance a Servlet may only provide the report data section of page while other servlets create the other components of the web page displayed to the user.

# 6. WDCT Classes

This chapter is an overview of all the classes used in WDCT. The WDCT classes are organised in a "flat" structure. This means that they are not contained in packages and have no hierarchical structure. This chapter complements the Javadoc documentation as discussed in section "3.1.6 Javadoc Directory" on page 5.

Each class below has a short description and a list of Session and Request parameters that it makes use of. The session parameters are the variables that contain information within the user's session object. The request parameters are passed to the Java class either via a HTTP Post message, which is accessed via the servlet's request object, or via the requested method's parameters.

### 6.1 AdministrationArea.class

AdministrationArea is a servlet that displays the Administration Area page with either only the reports or both the reports and site configuration tools depending on the privileges of the requesting administrator. This servlet is called after a correct user has logged in from the administrator login page.

#### Session Parameters:

username – Contains the username of the user logged into the system

### 6.2 AdministratorLogin.class

The AdministratorLogin class is a servlet that checks form data of a username and password against a list in the database before redirecting to the AdministrationArea servlet. If the login is incorrect then the user is redirected to administrator\_login\_error.jsp.

#### Request Parameters:

username – Contains the username of the user logged into the system

password – Contains the password that the user entered on the Administrator login page.

### 6.3 Apostrophe.class

The Apostrophe class is for fixing up apostrophe characters in strings that are used in JDBC queries. If a JDBC query contains a ' character it needs to be doubled up for storage in the database. The apostropheFIX(String) method performs this functionality. Currently Apostrophe class is used by creating an instance of it within other classes that are likely to use it. Alternatively Apostrophe could be made static and called directly when required although the system will need to be altered to accomplish this.

### 6.4 AssessmentAnalyticalReport.class

The AssessmentAnalyticalReport is a servlet that produces HTML pages for generating and displaying the Assessment Analytical Report. There are three pages that it can display the generate page, the report page and the print preview page. The generate page allows the user to select the criteria for displaying the report. The report page displays the report as to the criteria set on the generator page. The print preview page displays the report in a format conducive for printing. Not all the session variables are completed the first time that this Servlet is requested by a user as most of them are set after visiting the generator page that is also created by this Servlet.

#### Session Parameters:

- startDate – Contains the start date for the report period.
- endDate – Contains the end date for the report period.
- limit\_loc – Contains the name of the limiting location (optional).
- display\_name – Contains the name of a user to display data about (optional).
- groupby – Contains the name of the field to group the results by.
- firstorder – Contains the column name to order the results by first.
- Secondorder – Contains the column name to order the results by second.
- thirdorder – Contains the column name to order the results by third.

#### Request Parameters:

- Submit – Contains a value that indicates which page to be displayed, it indicates what button was pressed on each page. Expected values (null, Display, Back, Exit, Print Preview)
- demoid – Contains the demonstration's ID number so that the servlet knows which demonstration to display.
- StartD – Contains the start date for the report period.
- EndD – Contains the end date for the report period.
- limit\_loc – Contains the name of the limiting location (optional).
- display\_name – Contains the name of the user to display data about (optional).
- groupby – Contains the name of the column to group the results by.
- firstorder – Contains the name of the first column to order the results by.
- Secondorder – Contains the name of the second column to order the results by.
- Thirdorder – Contains the name of the third column to order the results by.

## 6.5 AssessmentStatusReport.class

`AssessmentStatusReport` is a servlet. It displays all the HTML pages needed for the assessment status report functionality. This includes the generator page, the report page, and the print preview page. The Assessment Status Report helps the user to understand how well the analysis process is progressing and displays a series of counters that describe how many days worth of analysis information has been gathered.

### Session Parameters:

- `startDate` – Contains the start date of the report period to be displayed in the report.
- `endDate` – Contains the end date of the report period to be displayed in the report.
- `response` – Contains the type of responses that should be displayed in the report. Expected values (`demonstrator`, `warfighter`).
- `order` – Contains name of the column to order the results by.
- `username` – Contains the name of a user currently logged into the system, which is displayed on the print version of the report.

### Request Parameters:

- `Submit` – Contains a value of the button that was pressed to load this Servlet. It is used to direct the logic of the Servlet. Expected values (`null`, `Display`, `Back`, `Exit`, `Print Preview`)
- `order` – expected values (`demo`, `loc`)
- `response` – Contains the type of responses to display in the report. Expected values (`demonstrator`, `warfighter`).
- `startD` – Contains the start date of the report time period to display.
- `endD` – Contains the end date of the report time period to display.

## 6.6 ConfigurationAdmin.class

This class displays the Configuration Admin page and most of the associated pages for the configuration administration tools. The tools included are "Change the Analysis period Start and End dates", "Change Contact Information" and "Change the Administrator Users".

### Session Parameters:

- `username` – Contains the username of the administrator currently logged in.
- `message` – Contains any error messages produced by the system for display to the user.

**Request Parameters:**

- Submit - Contains a value of the button that was pressed to load this Servlet. It is used to direct the logic of the Servlet. Expected values (null, Users, toAdvanced, toBasic, addAdvancedUser, addBasicUser, deleteBasic, deleteAdvanced, StartEnd, Update Dates, Contact, InsertContacts, Refresh, Back, Exit)
- BasicAdmin - Contains the name of the basic admin user that is selected.
- advancedUsername - Contains the new advanced user's username.
- advancedPassword - Contains the new advanced user's password.
- basicUsername - Contains the new basic user's username.
- basicPassword - Contains the new basic user's password.
- AdvancedAdmin - Contains the name of the advanced user that is selected.
- dropdowncolumn1 - Contains the start date of the demonstration period.
- dropdowncolumn2 - Contains the end date of the demonstration period.
- name1 - Contains the name of the primary site administrator.
- phone1 - Contains the phone number for the primary site administrator.
- mobile1 - Contains the mobile number for the primary site administrator.
- email1 - Contains the email address for the primary site administrator.
- name2 - Contains the name of the secondary site administrator.
- phone2 - Contains the phone number of the secondary site administrator.
- mobile2 - Contains the mobile number of the secondary site administrator.
- email2 - Contains the email address of the secondary site administrator.

**6.7 ConsentForm.class**

ConsentForm is a servlet that generates HTML necessary to display, edit, input and preview the consent form for the WDCT system. This class is also used for storing and retrieving the consent form information from the database.

**Request Parameters:**

- Submit - Contains the value of the button that was clicked to load this Servlet. Expected values are (null, Insert, Edit, Back).
- CFTitle - Contains the text for the consent form's title.
- CFBody - Contains the text for the consent form's body text.

**6.8 ContactInfo.class**

ContactInfo is a servlet used for generating contact information displayed at the bottom of most pages in the WDCT system. All contact information is stored within static variables and the contact information is output to the requesting servlet or JSP via the `display()` method. The `populate()` method gets the contact information from the database and places the information into the static variables.

## 6.9 CountDown.class

CountDown is a servlet that generates the JavaScript necessary for the client side session countdown functionality. The generated JavaScript should be included in all pages of the WDCT that require the session countdown clock functionality. Should the timeout ever be adjusted this is the class that needs to be modified.

## 6.10 CreateNvivoDocuments.class

CreateNvivoDocuments is a servlet that allows the user to output a compressed ZIP file containing a number of text documents in a format that QSR NVivo can understand. These reports are used when indepth analysis of the data is required.

### **Request Parameters:**

- |        |  |
|--------|--|
| Submit | - Contains the value of the button that was clicked to load this page, it is used to determine if the HTML page should be displayed or the ZIP file created. Expected values (null, notNull) |
|--------|--|

## 6.11 DBConnection.class

DBConnection is a servlet that contains information about the database connection and is used by other servlets to find out how to connect to the system database. Its primary role is to provide the address, username and password for the database. It retrieves this information from its initialisation parameters, which are stored in the context configuration file, web.xml.

## 6.12 DemonstrationAdmin.class

DemonstrationAdmin is a servlet that displays the Demonstration Administration Area page. From this area the administrator can insert, edit, view and delete demonstrations from the system.

### **Session Parameters**

- |          |   |
|----------|---|
| username | - Contains the username of the administrator user using the system. |
| demoid   | - Contains the number of the demonstration that is being altered.   |

### **Request Parameters**

- |             |  |
|-------------|--|
| Submit      | - Contains the name of the button that was clicked to display this page. The value is used to determine which logic within the Servlet should be executed. Expected values (null, Delete, YES, NO, Cancel, Exit, Edit, Display, New, Update, Insert) |
| demoid      | - Contains the ID number of the demonstration being altered.   |
| title       | - Contains the title of the demonstration being altered.   |
| description | - Contains the description of the demonstration being altered.   |

addedLocations – Contains a parameter value list (String array) of locations that have been added to the system.  
 deletedLocations – Contains a parameter value list (String array) of locations that have been deleted from the system.  
 selectedLocations – Contains a parameter value list (String array) of locations that have been selected for the demonstration.  
 warfighterSponsorList – Contains a list of warfighter sponsors via parameter values (String array). All the values (name, business phone, etc.) are stored as a single string for each warfighter sponsor; separated by a ‘;’.  
 defenceSponsorList – Contains a list of defence sponsors via parameter values (String array). All the values (name, business phone, etc.) are stored as a single string for each defence sponsor; separated by a ‘;’.  
 demonstrationSponsorList – Contains a list of demonstration sponsors via parameter values (String array). All the values (name, business phon, etc.) are stored as a single string for each demonstration sponsor; separated by a ‘;’.

### **6.13 Demonstrator.class**

Demonstrator is a servlet that displays the pages for a demonstrator to login into the system to enter a demonstrator log. The pages that this servlet produces are the “Choose Demonstration and Location” page and the “Choose Day” page.

#### **Session Parameters**

demoid	- Contains the ID number of the demonstration that the log will be entered into or the demonstration that logs will be retrieved from.
location	- Contains the name of the location that has been chosen to get demonstrator logs about or to enter new logs into.

#### **Request Parameters**

Submit	- Contains the name of the button clicked to display this page. This value is used to control the business logic flow within the servlet. Expected values (null, next, demonstration, location)
--------	---

### **6.14 DemonstratorLog.class**

DemonstratorLog is a servlet that displays the Demonstrator Log page, used to input information into demonstrator logs. This is one of the more basic servlets in the WDCT system. The page that is produced allows the user to view the currently inserted demonstrator logs or enter new logs.

**Session Parameters**

- |          |  |
|----------|--|
| date     | - Contains the date of the demonstrator log that are shown and will be entered into.                   |
| updated  | - Contains a flag for identifying if the log has been updated or not.                                  |
| demoid   | - Contains the ID number of the demonstrator that was selected in earlier pages.                       |
| location | - Contains the name of the location that the demonstrator log is being displayed from or entered into. |
| debug    | - Contains any debug information used by the servlet internally.                                       |

**Request Parameters**

- |        |   |
|--------|---|
| Submit | - Contains the name of the button that was clicked to display this page. This value is used to control the business flow within the Servlet. Expected values (null, Exit Without Update, Choose Demo, Exit) |
| date   | - Contains the date of the day that the log is being entered or retrieved about. It is not the data that the log was entered.   |
| answer | - Contains the text for the demonstrator log.   |

## **6.15 DemonstratorLogReport.class**

DemonstratorLogReport is a servlet that displays the pages associated with the Demonstrator Log Report. The pages that can be displays are the Generator page, the Report Page, and the Print Preview page. The report generated is essentially a list of the logs entered by demonstrators.

**Session Parameters**

- |           |   |
|-----------|---|
| startDate | - Contains the starting date for the period to be displayed in the report.      |
| endDate   | - Contains the end date for the period to be displayed in the report.           |
| order     | - Contains the name of the column to order the results by.                      |
| username  | - Contains the name of the username that is logged into the administrator area. |

**Request Parameters**

- |        |   |
|--------|---|
| Submit | - Contains the name of the button that was clicked to display this page. This value is used to control the business logic contained in this Servlet. Expected values (null, Display, Exit, Back, Print Preview) |
| order  | - Contains the name of the column used to order the results displayed in the report.  |
| startD | - Contains the starting date for the period to be displayed in the report.  |
| endD   | - Contains the end date for the period to be displayed in the report.   |

## 6.16 destroySession.class

destroySession is a servlet that destroys the session associated with the XMLHttpRequest object and thus the user. This is useful when you need to invalidate a session because of a security issue or when the session needs to be reset for other reasons. It should be kept in mind that there is a session timeout in place in WDCT that automatically will invalidate sessions after user inactivity. The session is nullified every time the user visits the site's home page.

## 6.17 EnterProfile.class

This servlet creates the "Profile" and "Consent" pages; it also inserts profiles into the database. The profiles are displayed when a warfighter logs into the system for the first time. It makes use of the ConsentForm servlet to provide the body of the consent form.

### Session Parameters

name	- Contains the name of the warfighter logging into the system.
rank	- Contains the rank of the warfighter logging into the system.
userid	- Contains the user ID number of the warfighter logging into the system.
service	- Contains the name of the service that the warfighter is working for.
debug	- Contains any debug information used by the system.

### Request Parameters

Submit	- Contains the value of the submit button that was clicked to display this page. This value is used to control the flow of business logic within this servlet. Expected values (null, yes, no)
currentunit	- Contains the value of the "Current Unit" field from the profile form.
currentpos	- Contains the value of the "Current Position" field from the profile form.
periodcurrentpos	- Contains the value of the "Period in Current Position" field from the profile form.
periodinADF	- Contains the value of the "Period in ADF" field from the profile form.
history	- Contains the value of the "Brief History" field from the profile form.
branch	- Contains the value of the "Branch" field from the profile form.

## 6.18 LessonsLearnt.class

LessonsLearnt is a servlet that displays the “Lessons Learnt Entry Form” and “Lessons Learnt List”, it also enters new lessons into the database.

### Session Parameters

- debug - Contains any debug information that is used by the system.

### Request Parameters

- |            |  |
|------------|--|
| Submit     | - Contains the value of the button that was clicked to display this page. Expected values (null, Cancel, List, Back, Insert, Exit) |
| lesson     | - Contains the value of the lesson being entered into the system.  |
| name       | - Contains the name of the user entering the lesson into the system.   |
| importance | - Contains the importance value as entered by the user for insertion into the system.  |

## 6.19 LessonsLearntReport.class

This servlet creates the pages for the Lessons Learnt Report. The pages produced are the “Lesson Learnt Report Generator” page, the “Lesson Learnt Report” page and the “Lesson Learnt Print Preview” page.

### Session Parameters

- |           |   |
|-----------|---|
| startDate | - Contains the starting date of the period to display in the report.      |
| endDate   | - Contains the end date of the period to display in the report.           |
| order     | - Contains the column name to order the results by.                       |
| username  | - Contains the username of the administrator user logged into the system. |

### Request Parameters

- |        |   |
|--------|---|
| Submit | - Contains the value of the button that was clicked to display this Servlet. This value is used to control the business logic flow within the Servlet. Expected values (null, Display, Exit, Back, Print Preview) |
| order  | - Contains the column name to order the results by for the report.  |
| startD | - Contains the start date for the period to display in the report.  |
| endD   | - Contains the end date for the period to display in the report.  |

## 6.20 NetworkAdmin.class

NetworkAdmin is a servlet that displays the login pages to access the network administrators log. The two pages that can be displayed by this servlet are the Choose Location page and the Choose Day page.

**Session Parameters**

- location - Contains the text string of the location selected by the network administrator.

**Request Parameters**

- Submit - Contains the value of the button that was clicked to display this Servlet. The value is used to control the business logic flow contained within the Servlet. Expected values (null, Next)
- location - Contains the name of the location that any new network admin logs will be saved at.

**6.21 NetworkAdminLog.class**

`NetworkAdminLog` is a servlet that displays the network administrator log to the user and also insert new network admin logs into the database.

**Session Parameters**

- date - Contains the date of the Network Admin Log to display
- location - Contains the name of the location to display Network Admin Logs about.
- debug - Contains any debug information that is used by the system.

**Request Parameters**

- submit - Contains the value of the button that was clicked to the display this Servlet. This value is used to control the business logic contained within the Servlet. Expected values (null, Cancel, Insert)
- answer - Contains the Network Admin Log text to be inserted into a new Network Admin Log.
- name - Contains the name of the person who has entered the new Network Admin Log.
- contact - Contains the contact number of the person who has entered the new Network Admin Log.

**6.22 NetworkPerformanceReport.class**

`NetworkPerformanceReport` is a servlet that produces the pages for the Network Performance report. The pages include the "Network Performance Report Generator" page, the "Network Performance Report" page, and the "Network Performance Report Print Preview" page.

**Session Parameters**

- startDate - Contains the start date of the result period to display Network Logs about.
- endDate - Contains the end date of the results period to display Network Logs about.

debugStrSQL - Contains debug SQL strings for use internally within the system.  
 username - Contains the name of the user who is logged into the Administrator area.

#### Request Parameters

Show	- Contains the value of the button clicked to display this Servlet. This value is used to control the business logic flow within the Servlet. Expected values (null, Exit, Back, Print, Report)
startD	- Contains the start date of the results period to display in the report.
endD	- Contains the end date of the results period to display in the report.

#### 6.23 QuestionAdmin.class

QuestionAdmin is a servlet that allows an administrator to insert/edit/delete questions in the system. A great part of the functionality that this servlet provides is supplemented with JavaScript. The way that some actions are performed on questions could be improved, for instance editing a question doesn't require the question to be deleted and then recreated.

#### Session Parameters

username	- Contains the username of the Administrator that is logged in.
demoid	- Contains the ID number of the demonstration that is being edited.
question_no	- Contains the number of the question being edited.
sub_question_no	- Contains the sub question number of the question being edited.
type	- Contains the type of the question being edited.
message	- Contains any string messages that are to be displayed to the user.
category	- Contains the category string for the question being edited.
question	- Contains the question text for the question that is being edited.
questionid	- Contains the ID number of the question being edited.
arguments	- Contains the arguments 1 value for the question being edited.
arguments2	- Contains the arguments 2 value for the question being edited.
choices	- Contains the choices value for the question being edited.
title	- Contains the title for the question that is being edited.

**Request Parameters**

Submit - Contains the value of the button that was clicked to display this Servlet. This value is used to control the business process flow within the Servlet. Expected values (null, List, Next, Cancel, Insert Question, Insert Main Question, Insert Sub Question, Insert Coding, move up, Move Down, to main, to sub, Complete Conversion, Free Text, Rating Scale, Yes / No, Multiple Choice, Insert, Update, Change Demo, Delete, Edit, Exit)

demonstration - Contains the ID number of the demonstration that is being edited.

selectedQuestionNo - Contains the question number of the selected question.

selectedSubQuestionNo - Contains the sub question number of the selected question.

insertChk - Contains the value of whether a new question should be inserted before or after the currently selected question.

question\_no - Contains the question number of the question being inserted or edited.

sub\_question\_no - Contains the sub question number of the question being inserted or edited.

coding - Contains the coding information of the question being inserted or edited.

category - Contains the category text of the question being inserted or edited.

question\_text - Contains the question text of the question being inserted or edited.

arguments - Contains the arguments 1 value for the question being inserted or edited.

arguments2 - Contains the arguments 2 value for the question being inserted or edited.

type - Contains the question type for the question being inserted or edited.

choices - Contains a parameter list (String array) of the choices for the question being inserted or edited.

SelectedQuestionID - Contains the ID number of the selected question.

**6.24 Questions.class**

Questions is a servlet that displays the questions to the warfighter within the questions window in the Warfighter Area. This is probably the most used servlet in the entire WDCT system. During development this was the source of the most problems. Be very careful when modifying this Servlet.

#### Session Parameters

question\_no - Contains the question number of the question to be displayed.  
sub\_question\_no - Contains the sub question number of the question to be displayed.  
demoid - Contains the ID number of the demonstration that the questions being displayed are from.  
date - Contains the date that the warfighter is entering data for.  
userid - Contains the user ID number for the warfighter that is answering the questions.  
location - Contains the name of the location that the warfighter is entering in answers for.

#### Request Parameters

Submit - Contains the value of the button that was clicked to display this servlet. This value is used to control the flow of business logic contained in the Servlet. Expected values (null, Exit, Goto, Previous, endPrevious, Next.)  
QuestionNo - Contains the question number of the question that was just displayed.  
SubQuestionNo - Contains the sub question number of the question that was just displayed.  
answer - Contains the answer that the warfighter entered for the last question displayed.

### 6.25 UsersReport.class

UserReport is a servlet that creates the pages for the User report. The pages included are the "Users Generator" page, the "Users Report" page and the "User Print Preview" page.

#### Session Parameters

order - Contains the column name to order the reports results by.  
debugStrSQL - Contains debug information that is used within the system.  
username - Contains the administrator user's username.

#### Request Parameters

Submit - Contains the value of the button clicked to display this Servlet. This value is used to control the business logic flow within the Servlet. Expected values (null, Display, Exit, Back, Print Preview)  
order - Contains the column name to order the reports results by.

## 6.26 Warfighter.class

Warfighter is a servlet that allows a warfighter to log into the system by displaying the "Choose Demonstration and Location" page and the "Choose Day" page.

### Session Parameters

- |          |   |
|----------|---|
| demoid   | - Contains the ID number of the demonstration selected by the warfighter. |
| location | - Contains the name of the location selected by the warfighter.           |
| userid   | - Contains the user ID number of the warfighter logged into the system.   |

### Request Parameters

- |               |  |
|---------------|--|
| Submit        | - Contains the value of the button clicked to display this Servlet. This value is used to control the business logic flow within the Servlet. Expected values (null, Next) |
| demonstration | - Contains the ID number of the demonstration that was selected.   |
| location      | - Contains the name of the location selected.  |

## 6.27 WarfighterLog.class

WarfighterLog is a servlet that displays the warfighter log in the warfighter log window.

### Session Parameters

- |          |  |
|----------|--|
| date     | - Contains the date that the warfighter log will be retrieved from or inserted to. |
| userid   | - Contains the user ID of the warfighter that is logged into the system.           |
| demoid   | - Contains the ID number of the demonstration that was selected by the warfighter. |
| location | - Contains the location that was selected by the warifghter.                       |

### Request Parameters

- |        |   |
|--------|---|
| Submit | - Contains the value of the button that was clicked to display this Servlet. This value is used to control the business logic flow within the Servlet. Expected values (null, Cancel, Update) |
| answer | - Contains the answer entered by the warfighter to be inserted into the current warfighter log.   |

## 6.28 WarfighterLogin.class

WarfighterLogin is a servlet that checks the database for an instance of the user in the database. If there is no entry for the user then their browser is redirected to the EnterProfile Servlet for their account to be created. If an entry is found then their details are loaded into the user session and the browser is redirected to the Warfighter Servlet.

### Session Parameters

- |         |   |
|---------|---|
| name    | - Contains the name of the warfighter as entered into the "Name" field on the "Warfighter Login" page.          |
| service | - Contains the service the warfighter is in as entered into the "Service" field on the "Warfighter Login" page. |
| rank    | - Contains the rank of the warfighter as entered into the "Rank" field on the "Warfighter Login" page.          |
| userid  | - Contains the user ID number of the logged in warfighter.  |

### Request Parameters

- |          |   |
|----------|---|
| username | - Contains the name of the warfighter as entered into the "Name" field on the "Warfighter Login" page.          |
| rank     | - Contains the rank of the warfighter as entered into the "Rank" field on the "Warfighter Login" page.          |
| service  | - Contains the service the warfighter is in as entered into the "Service" field on the "Warfighter Login" page. |

## 6.29 warfighter\_split.class

warfighter\_split is a servlet that displays the "Warfighter Split" page; from this page the warfighter can open the "Questions" window or the "Warfighter Log" window to enter analysis information into the database.

### Session Parameters

- |          |   |
|----------|---|
| date     | - Contains the date selected for the entering of analysis information.    |
| location | - Contains the name of the location selected by the warfighter.           |
| demoid   | - Contains the ID number of the demonstration selected by the warfighter. |

### Request Parameters

- |      |  |
|------|--|
| date | - Contains the date selected by the warfighter on the "Choose Day" page. |
|------|--|

## 7. WDCT Java Server Pages (JSP)

This chapter is a short summary of the JSPs used in the WDCT system.

### 7.1 administrator\_login.jsp

Provides the administrator login functionality. The Servlet performs no input validity checks. The entered information into the two fields is sent directly to the AdministratorLogin Servlet.

### 7.2 administrator\_login\_error.jsp

This JSP is very similar to the administrator\_login.jsp, except it informs the user that their previous login attempt was incorrect.

### 7.3 db\_error.jsp

This JSP is displayed when a database connection error is detected. This is usually when a database connection cannot be established. Output generated by this JSP informs the user of the problem and instructs them to contact the system administrator.

### 7.4 index.jsp

This is the initial page seen by the user when they first enter the WDCT system. This page provides links to all other sections. It also invalidates any sessions that the user might have had before on in this system.

### 7.5 invalid\_session.jsp

This page is displayed when an invalid session is detected. A session may become invalid due to a timeout, or because a user has no access to a certain area with their current session. An example of the latter is a user who tries to access the report servlets from the administration area directly without following the administrator login procedure. This situation will not create the required session object and therefore the report Servlet will not display correctly and the "Invalid Session" page is displayed.

### 7.6 warfighter\_login.jsp

This JSP displays the warfighter login screen. It enables the warfighter to submit their name, service and rank. The rank selection menu is JavaScript driven and displays only the ranks available for the selected service.

## 8. WDCT Database Schema

This chapter is a summary of the WDCT database schema. Listed are all the tables and their columns. Column names in bold type and a "\*" at the end of their name are primary key columns for the table. The WDCT database schema also contains foreign key references between tables, which are not recorded below. However column names that are identical between tables are usually a foreign key reference. Also contained in the database schema are sequences that are used to increment ID numbers throughout the system. One example is for the demonstration ID number. For the full database schema consult `create_db.sql` file on this WDCT Developer CD-ROM.

### 8.1 ADMINISTRATORS Table

#### 8.1.1 Columns

<b>USERNAME*</b>	Contains the username of the administrator user.
<b>PASSWORD*</b>	Contains the clear text password of the administrator user.
<b>SITE_ADMIN_PRIV</b>	Contains the administrator users priviledges (either 'y' or 'n')

#### 8.1.2 Description

This table is used to store all the information about the administrator users contained within the system. This table is checked when users log into the administrator area. The "Admin Users" page also manipulates and sets up the users contained in this table. When the database is first created this table will have a default entry of:

USERNAME:	default
PASSWORD:	default
SITE_ADMIN_PRIV:	y

### 8.2 ANSWERS Table

#### 8.2.1 Columns

<b>QUESTIONID*</b>	Contains the ID number of the question that has been answered.
<b>USERID*</b>	Contains the user ID number of the warfighter who entered this answer.
<b>LOCATION*</b>	Contains the location that this answer is attached to.
<b>DAY*</b>	Contains the day that this answer is attached to.
<b>ANSWER*</b>	Contains the answer text entered by the warfighter.
<b>DATE_ANSWERED*</b>	Contains the date that the warfighter entered this answer.
<b>DEMOID*</b>	Contains the demonstration ID number that this answer is attached to.

### 8.2.2 Description

This table is used to store the answer entered by the warfighters. Each answer is stored with a reference to the location, day, and demonstration that it is associated with. The question ID column identifies which question was answered. The Questions Servlet uses this table to store information entered by the users of its pages.

## 8.3 DEFENCE\_SPONSOR Table

### 8.3.1 Columns

<b>DEMOID*</b>	Contains the Demonstration ID number that the sponsor is associated to.
NAME	Contains the name of the Defence Sponsor.
BUSINESS_PHONE	Contains the business phone number of the Defence Sponsor.
MOBILE_PHONE	Contains the mobile phone number of the Defence Sponsor.
FAX	Contains the fax number of the Defence Sponsor.
EMAIL	Contains the email address of the Defence Sponsor.
POSTAL_ADDRESS	Contains the postal address of the Defence Sponsor.

### 8.3.2 Description

This table contains all the Defence Sponsors associated with each of the demonstrations within the system. The DemonstrationAdmin Servlet uses this table to view and manipulate the Defence Sponsors for each demonstration.

## 8.4 DEMONSTRATIONS Table

### 8.4.1 Columns

<b>DEMOID*</b>	Contains the Demonstration ID number for each demonstration in the system. This number is unique for each demonstration.
TITLE	Contains a short title of the demonstration.
DESCRIPTION	Contains a long description of what the demonstration is about and what it is used for.

### 8.4.2 Description

This is the core table to store information about demonstration contained in the system. All the demonstrations will have one entry within this table. This table is also used to find the demonstration ID number for a particular demonstration. Numerous pages within the system use it. When inserting a new record into this table a new demonstration ID number must be generated using the Demonstration ID database sequence (see the `create_db.sql` file for the name of the sequence)

## 8.5 DEMONSTRATION\_SPONSOR Table

### 8.5.1 Columns

<b>DEMOID*</b>	Contains the name of the demonstration that the demonstration sponsor is associated with.
NAME	Contains the name of the demonstration sponsor point of contact.
BUSINESS_PHONE	Contains the phone number of the demonstration sponsor point of contact.
MOBILE_PHONE	Contains the mobile phone number of the demonstration sponsor point of contact.
FAX	Contains the fax number of the demonstration sponsor point of contact.
EMAIL	Contains the email address of the demonstration sponsor point of contact.
POSTAL_ADDRESS	Contains the postal address of the demonstration sponsor point of contact.

### 8.5.2 Description

This table contains information about the demonstration sponsors that are attached to each of the demonstrations. The DemonstrationAdmin Servlet uses this table to retrieving and inserting demonstration sponsors. The demonstration ID stored within this table must be an existing demonstration ID stored within the DEMONSTRATIONS table.

## 8.6 DEMONSTRATOR\_LOG Table

### 8.6.1 Columns

<b>DEMOID*</b>	Contains the demonstration ID number for the demonstration that this log is about.
<b>LOCATION*</b>	Contains the location that this log was entered for.
<b>DAY*</b>	Contains the day that this log is to be associated with.
LOG	Contains the text of the log entered.

### 8.6.2 Description

This table contains all the information required to store demonstrator logs within the system. The DemonstratorLog and DemonstratorLogReport Servlets use it for the entering and displaying of demonstrator logs.

## 8.7 DEMO\_LOCATIONS Table

### 8.7.1 Columns

<b>DEMOID*</b>	Contains the ID number of the demonstration.
<b>LOCATION*</b>	Contains the name of the location (eg. Sydney).
POC_NAME	Contains the name of the point of contact at this location.
POC_BUSINESS_PHONE	Contains the business phone number of the point of contact at this location.
POC_MOBILE_PHONE	Contains the mobile phone number of the point of contact at this location.
POC_FAX	Contains the fax number of the point of contact at this location.
POC_EMAIL	Contains the email address of the point of contact at this location.
POC_POSTAL_ADDRESS	Contains the postal address of the point of contact at this location.
COMMENTS	Contains any comments about this location for this demonstration.

### 8.7.2 Description

Currently only the DEMOID and LOCATION columns are being used in the database. All the other fields support future functionality that demonstration point of contacts could be added for each demonstration location. Overall this table is used to link the demonstration with one or more locations.

## 8.8 JWID\_ADMIN Table

### 8.8.1 Columns

START_DATE	Contains the start date for the overall demonstration period.
END_DATE	Contains the end date for the overall demonstration period.
YEAR	Contains the year that the demonstrations are being run.
CONTACT1_NAME	The name of the first site administrator contact.
CONTACT1_EMAIL	The email of the first site administrator contact.
CONTACT1_PHONE	The phone number of the first site administrator contact.
CONTACT1_MOBILE	The mobile number of the first site administrator contact.
CONTACT2_NAME	The name of the second site administrator contact.
CONTACT2_EMAIL	The email address of the second site administrator contact.
CONTACT2_PHONE	The phone number of the second site administrator contact.
CONTACT2_MOBILE	The mobile number of the second site administrator contact.

### 8.8.2 Description

This table will only contain one role and has no primary keys. This table is useful throughout the system for basic site configuration information. Default information is entered when the database is created. The ContactInfo Servlet uses this page to display the contact information at the bottom of all the sites pages. The ConfigurationAdmin Servlet also uses this table to alter the contact information and analysis period.

## 8.9 LESSONSLEARNT Table

### 8.9.1 Columns

LESSON	Contains the text of the lesson learnt entered by the user.
NAME	The name of the user who entered the lesson learnt entry.
ENTERED_ON	The date that the lesson learnt was entered (based on the servers operating system date/time).
IMPORTANCE	The importance level of the lesson learnt as entered by the user.

### 8.9.2 Description

This table is used by the lesson learnt capability within the WDCT system to store and retrieve new or existing lessons learnt.

## 8.10 LOCATION\_LOV Table

### 8.10.1 Columns

LOCATION	Contains the location name.
COMMENTS	Contains any comments about the location.

### 8.10.2 Description

This table contains a complete list of locations used within the system. Currently the COMMENTS column is not used by any component of the system but is reserved for future capability.

## 8.11 MULTIPLE\_CHOICE Table

### 8.11.1 Columns

QUESTIONID*	The question ID number of the question that the choice is attached to.
CHOICE_TEXT*	The text of the actual choice.
ORDER_NO	The order number the choice is in a list (eg. 1 = first place)

### 8.11.2 Description

This table contains all the multiple choice options for each of the multiple choice questions within the system. For each multiple choice type question in the system there will be at least one entry in this table for the same QUESTIONID. The order number determines where in the list does the choice get displayed.

## 8.12 NETWORK\_ADMIN\_LOG Table

### 8.12.1 Columns

<b>LOCATION*</b>	Contains the name of the location that the network admin log is associated with.
<b>DAY*</b>	The day that the network admin log is about.
<b>LOG*</b>	Contains the body text for the log itself.
<b>DATE_ENTERED*</b>	The date that the log was entered into the system.
<b>NAME*</b>	The name of the user who entered the log.
<b>CONTACT_NUMBER*</b>	The contact number of the user who entered the log.

### 8.12.2 Description

This table is used to store all the network administration logs that are entered into the system by the NetworkAdminLog Servlet. All the fields in this table are mandatory.

## 8.13 PROFILES Table

### 8.13.1 Columns

<b>USERID*</b>	The user ID number for the warfighter.
DATE_ENTERED	The date that the user account was created.
CURRENT_UNIT	The unit that the warfighter is currently in.
CURRENT_POS	The current position the warfighter is in.
PERIOD_IN_CURRENT_POS	Contains the time that the warfighter has been in their current position.
PERIOD_IN_ADF	The time the warfighter has spent in the Australian Defence Force.
CAREER_HISTORY	Contains a brief history about the warfighter.
CURRENT_BRANCH	The branch that the warfighter is currently in.
RANK	The warfighter's rank.
SERVICE	The service that the warfighter works for.

### 8.13.2 Description

This table is used to store all the warfighter profiles. These table entries are used to find out more information about users. The usernames are not stored in this table as names should not be directly related to the answers entered into the system and because the userid is used in ANSWERS table it was decided not to include usernames in this table.

## 8.14 QUESTIONS Table

### 8.14.1 Columns

<b>QUESTIONID*</b>	The question's ID number. Used when referencing a question within the system.
DEMOID	The ID number of the demonstration that this question is attached to.
TYPE	The type of this question.
QUESTION	Contains the question text that is displayed to the warfighter.
QUESTION_NO	The questions number.
SUB_QUESTION_NO	The question's sub question number. (eg. This would contain the value 3 if the questions number was 10.3)
CATEGORY	The category text to display at the top of the question when a warfighter is answering the question.
ARGUMENTS	Contains extra information about the question. This is used for some question types.
ARGUMENTS2	This is used for extra information when the ARGUMENTS column is already used. This is used when two extra values are required for a question.

### 8.14.2 Description

This table contains all the questions that are displayed to the warfighter on the Questions page. The QUESTION\_NO and SUB\_QUESTION\_NO are used to determine where the question is in the list. The TYPE column determines what type of question is to be displayed and therefore which HTML should be created to display the question.

## 8.15 USERS Table

### 8.15.1 Columns

<b>USERID*</b>	Contains the users ID number.
NAME	Contains the name of the warfighter.
RANK	Contains the rank of the warfighter.

### 8.15.2 Description

This table is used to log in warfighters into the "Warfighter Area". When the user enters their name, service and rank onto the "Warfighter Login" page then these values are checked against this table. If the values entered do not match the entries in the table then it is assumed that this is a new user logging into the system and therefore displays the "Consent Form" and "Profile" pages. When new users are entered into the system both this USERS table and the PROFILES table must be updated.

## 8.16 WARFIGHTER\_LOG Table

### 8.16.1 Columns

<b>USERID*</b>	Contains the warfighters user ID number.
<b>DEMOID*</b>	The demonstration ID number that the log is associated with.
<b>LOCATION*</b>	The location that the log is associated with.
<b>DAY*</b>	The day that the log is associated with.
<b>LOG</b>	Contains the complete text inserted into the log entry.

### 8.16.2 Description

This table contains all the warfighter log entries. Each log is associated with a particular demonstration, location and day. The WarfighterLog, AssessmentAnalyticalReport, AssessmentStatusReport and CreateNvivoDocuments Servlets use this table.

## 8.17 WARFIGHTER\_SPONSOR Table

### 8.17.1 Columns

DEMOID	Contains the demonstration ID number that the warfighter sponsor is attached to.
NAME	The name of the warfighter sponsor point of contact.
BUSINESS_PHONE	The business phone of the warfighter sponsor point of contact.
MOBILE_PHONE	The mobile phone of the warfighter sponsor point of contact.
FAX	The fax number of the warfighter sponsor point of contact.
EMAIL	The email address of the warfighter sponsor point of contact.
POSTAL_ADDRESS	The postal address of the warfighter sponsor point of contact.

### 8.17.2 Description

This table contains all the warfighter sponsors that are attached to each of the demonstrations. This is viewed and edited through the DemonstrationAdmin Servlet.

## 9. Future Directions

The WDCT system is fully functional; however there are a number of improvements that could be made if the opportunity arises. Below is a list of the improvements that could be made.

### 9.1 Database Connection Pool

#### Why

Currently the implemented Database communication throughout WDCT is not structured. Every time database communication is required either a new connection is opened or a servlet wide connection is used. This approach is not very efficient and can cause problems if large numbers of connections are required by the WDCT system simultaneously. A better solution to the current approach is to use a database connection pool. Such a pool manages the database connections, and when one is required it is used by the system component. Once the system component finishes using the DB connection it is returned to the connection pool. This way, there is a lot less overhead associated with opening and closing DB connections as they are reused. This approach also alleviates problems with connections idle timing out.

#### Work required

Implementation of a DB connection pool is not a difficult task. There are commercial and open source implementations of database connection pools that could be made use of. The work intensive task in implementing this in the WDCT system is the fact that ALL the current database communication code would need to be rewritten to make use of the connection pool. Since database communication is used heavily throughout the WDCT, this is quite a substantial task.

### 9.2 User Input Checking and Verification

#### Why

There are many areas of the WDCT that do not perform checks on user input. Other areas do not check for all aspects of input that could potentially break the WDCT. This would not be an issue if users could be expected to provide correct input every time. Because WDCT users are human, perfect input cannot be expected. Thus the WDCT is susceptible to failure or incorrect behaviour due to incorrect user input.

#### Work Required

User input throughout the WDCT should be checked for invalid characters. Such characters could cause data to be not stored or break some system functionality.

User input should also be checked for relevance. Lastly user input length should be checked to make sure that it can be stored in the space allocated for this input type in the database.

### 9.3 Browser Compatibility

#### Why

Currently the WDCT only supports one browser, the MS Internet Explorer. This limits the users greatly, as this browser is not available on all computing platforms. Making the WDCT compatible with different browsers would increase the overall use and flexibility.

#### Work Required

Because WDCT relies on JavaScript and DHTML it is difficult to make it compatible with all browsers. Further some browsers do not adhere to some HTML and Internet standards. Therefore, to make the WDCT compatible with many browsers, potentially many implementations of the same functionality would need to be made.

### 9.4 Role Based Administrator system

#### Why

Currently there are only two types of administrator users. This is not always sufficient to provide the right level of privileges.

#### Work Required

Creation of more flexible administrator user system is a moderately complicated task. The administration area would have to be modified quite significantly to make such functionality possible. The administrator management area would have to undergo a total rewrite. The database structure would also be required to change for the storage of these added priviledges or roles.

### 9.5 More Sophisticated Question System

#### Why

The question system in WDCT is quite extensive but not flexible. The users are restricted to the question types that already exist in the system. Further the sub question functionality only allows for one level of hierarchy. These factors may sometimes limit how the data is collected and lead to a less efficient data collection process.

#### Work Required

To make the WDCT question system more flexible, its design would have to be changed. This means that it would have to be rewritten from ground up, with minimal code reuse from the current implementation.

## 9.6 Export Functionality

### Why

The system requires greater functionality to export the analysis information into formats understood by other analysis packages (eg. Microsoft Excel). Although the system already supports output to NVivo this support is only very basic.

### Work Required

This is a fairly minor change by adding this extra functionality directly to the Administration area. The complexity of creating this functionality really depends on the complexity of the format that is being exported to. For example creating an XML output function would be fairly simple, while an export to Microsoft Excel could be quite challenging.

## 10. Conclusion

Overall the WDCT system is a very basic Servlet and JSP implementation within the Jakarta Tomcat server. Since its development there have been many advances in technology and standards that could now be used within the WDCT. For instance the use of portal technology could help by the creation of user roles and depending on the users role would determine which parts of the WDCT system would be displayed. Hopefully this system will help you in your software demonstration analysis. With new iterations of this system it will become more capable and flexible.

## References

1. Hunter, J., Crawford, W. (2001) *Java Servlet Programming, 2<sup>nd</sup> Edition*, O'Reilly & Associates Inc.
2. Bergsten, H. (2002) *JavaServer Pages, 2<sup>nd</sup> Edition*, O'Reilly & Associates Inc.
3. Kuster, E. (2003) *Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) User Guide*, DSTO Command & Control Division
4. Kuster, E. (2003) *Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Installation Manual*, DSTO Command & Control Division

## DISTRIBUTION LIST

DSTO-GD-0377

Joint Warrior Interoperability Demonstration (JWID)  
Web Data Collection Tool (WDCT)  
Developer Guide

*Egon Kuster*

## AUSTRALIA

### DEFENCE ORGANISATION

	<b>No. of copies</b>
<b>Task Sponsor</b>	
Deputy Director JWID (CMDR Stevan Tomkins , R1-3-A079)	1
Deputy Director Combined & Coalition Information Networks (Mr Andrew Tape R1-3-A112)	1
<b>S&amp;T Program</b>	
Chief Defence Scientist	
FAS Science Policy	shared copy
AS Science Corporate Management	
Director General Science Policy Development	
Counsellor Defence Science, London	Doc Data Sheet
Counsellor Defence Science, Washington	Doc Data Sheet
Scientific Adviser to MRDC, Thailand	Doc Data Sheet
Scientific Adviser Joint	1
Navy Scientific Adviser	Doc Data Sheet & Dist List
Scientific Adviser - Army	Doc Data Sheet & Dist List
Air Force Scientific Adviser	Doc Data Sheet & Dist List
Scientific Adviser to the DMO M&A	Doc Data Sheet & Dist List
Scientific Adviser to the DMO ELL	Doc Data Sheet & Dist List
Director of Trials	1
<b>Information Sciences Laboratory</b>	
Chief Command & Control Division	Doc Data Sheet
Research Leader Command & Intelligence Environments Branch	1
Research Leader Military Information Enterprise Branch	1
Research Leader Theatre Operations Analysis Branch	1
Head Virtual Enterprises	Doc Data Sheet
Head Systems Simulation and Assessment	Doc Data Sheet
Head Theatre Operations Analysis	Doc Data Sheet
Head Intelligence Analysis	Doc Data Sheet
Head Human Systems Integration	Doc Data Sheet
Head C2 Australian Theatre	Doc Data Sheet
Head HQ Systems Experimentation	1
Head Information Systems	Doc Data Sheet
Head Information Exploitation	Doc Data Sheet
Egon Kuster (Author)	2
Publications and Publicity Officer, C2D/EOC2D	1 shared copy

<b>DSTO Library and Archives</b>		
Library Edinburgh	1 copy & Doc Data Sheet	
Australian Archives	1	
Library Canberra	Doc Data Sheet	
 <b>Capability Systems Division</b>		
Director General Maritime Development	Doc Data Sheet	
Director General Land Development	1	
Director General Aerospace Development	Doc Data Sheet	
Director General Information Capability Development	Doc Data Sheet	
 <b>Office of the Chief Information Officer</b>		
Deputy CIO	Doc Data Sheet	
Director General Information Policy and Plans	Doc Data Sheet	
AS Information Structures and Futures	Doc Data Sheet	
AS Information Architecture and Management	Doc Data Sheet	
Director General Australian Defence Simulation Office	Doc Data Sheet	
 <b>Strategy Group</b>		
Director General Military Strategy	Doc Data Sheet	
Director General Preparedness	Doc Data Sheet	
 <b>HQAST</b>		
SO (Science) (ASJIC)	Doc Data Sheet	
 <b>Navy</b>		
Director General Navy Capability, Performance and Plans, Navy Headquarters	Doc Data Sheet	
Director General Navy Strategic Policy and Futures, Navy Headquarters	Doc Data Sheet	
 <b>Army</b>		
ABC National Standardisation Officer, Land Warfare Development Sector, Puckapunyal	e-mailed Doc Data Sheet	
SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), Enoggera QLD	Doc Data Sheet	
SO (Science) - Land Headquarters (LHQ), Victoria Barracks NSW	Doc Data & Exec Summary	
 <b>Intelligence Program</b>		
DGSTA Defence Intelligence Organisation	1	
Manager, Information Centre, Defence Intelligence Organisation	1	
Assistant Secretary Corporate, Defence Imagery and Geospatial Organisation	Doc Data Sheet	
 <b>Defence Materiel Organisation</b>		
Head Airborne Surveillance and Control	Doc Data Sheet	
Head Aerospace Systems Division	Doc Data Sheet	
Head Electronic Systems Division	Doc Data Sheet	
Head Maritime Systems Division	Doc Data Sheet	

Head Land Systems Division	Doc Data Sheet
Head Industry Division	Doc Data Sheet
Chief Joint Logistics Command	Doc Data Sheet
Management Information Systems Division	Doc Data Sheet
Head Materiel Finance	Doc Data Sheet

#### **Defence Libraries**

Library Manager, DLS-Canberra	Doc Data Sheet
Library Manager, DLS - Sydney West	Doc Data Sheet

#### **OTHER ORGANISATIONS**

National Library of Australia	1
NASA (Canberra)	1

#### **UNIVERSITIES AND COLLEGES**

Australian Defence Force Academy	
Library	1
Head of Aerospace and Mechanical Engineering	1
Hargrave Library, Monash University	Doc Data Sheet
Librarian, Flinders University	1

### **OUTSIDE AUSTRALIA**

#### **INTERNATIONAL DEFENCE INFORMATION CENTRES**

US Defense Technical Information Center	2
UK Defence Research Information Centre	2
Canada Defence Scientific Information Service	e-mail link to pdf
NZ Defence Information Centre	1

#### **ABSTRACTING AND INFORMATION ORGANISATIONS**

Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts, US	1
Documents Librarian, The Center for Research Libraries, US	1

SPARES 5

**Total number of copies:** 35

<b>DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA</b>		1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	
<b>2. TITLE</b> Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Developer Guide		<b>3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)</b> Document (U) Title (U) Abstract (U)	
<b>4. AUTHOR(S)</b> Egon Kuster		<b>5. CORPORATE AUTHOR</b> Information Sciences Laboratory PO Box 1500 Edinburgh South Australia 5111 Australia	
<b>6a. DSTO NUMBER</b> DSTO-GD-0377		<b>6b. AR NUMBER</b> AR-012-920	
<b>8. FILE NUMBER</b> 9505-23-35		<b>9. TASK NUMBER</b> 01/307	
<b>10. TASK SPONSOR</b> DDJWID		<b>11. NO. OF PAGES</b> 46	
<b>13. URL on the World Wide Web</b> <a href="http://www.dsto.defence.gov.au/corporate/reports/DSTO-GD-0377.pdf">http://www.dsto.defence.gov.au/corporate/reports/DSTO-GD-0377.pdf</a>		<b>14. RELEASE AUTHORITY</b> Chief, Command and Control Division	
<b>15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT</b>  <i>Approved for public release</i>			
OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111			
<b>16. DELIBERATE ANNOUNCEMENT</b> No Limitations			
<b>17. CASUAL ANNOUNCEMENT</b> Yes			
<b>18. DEFTEST DESCRIPTORS</b> Joint Operations, Interoperability, Demonstrations, Data Analysis, Data Recording, Software Tools			
<b>19. ABSTRACT</b> This document has been created for the developer of the WDCT system. It contains documentation of the WDCT architecture and components that are part of the system. To support this document is the WDCT Developer CD-ROM, which contains all the WDCT source files and support files that are required in the development or alteration of the system. The structure of the database and the data contained within is explained. All Java Class files are discussed as to their function within the WDCT. Each JSP used is described as to the functionality that it provides. Also contained is a brief discussion of what new functionality that could be added.			